

# **CONTROL OF COMPLEX INTEGRATED AUTOMATED SYSTEMS – SYSTEM RETRO-FIT WITH AGENT-BASED TECHNOLOGIES AND INDUSTRIAL CASE EXPERIENCES**

Dr. Robert J. Graves

Dr. Valika K. Wan  
Thayer School of Engineering  
Dartmouth College

Mr. Jan van der Velden

Mr. Bruno van Wijngaarden  
Vanderlande Industries, The Netherlands

## **Abstract**

This paper addresses trends in control system development, especially agent-based approaches, which offer benefits over classical rule-based approaches. What are these, how do they work in practice and what performance benefits might result from them are topics addressed. Computational results developed from a research relationship with Vanderlande Industries (The Netherlands) are reported for a full scale Order Distribution System application of a goods-to-operator system.

## **1 Introduction**

Modern manufacturing systems often operate within environments where various forms of complexities and uncertainties exist [8], [5]. Traditional control systems, such as those constructed using hierarchical control architectures, follow a “top-down” structure characterized by “parent-child” relationships. They rely on a static set of assumptions and are intrinsically inflexible when reacting to unexpected events [16]. This explains why traditional hierarchical control architectures may not adequately address the needs of a complex system [7], [1], [4]. An inbuilt set of assumptions about certain environmental conditions and a prescribed set of options for system operation can become outmoded quickly. Any minor fluctuation in external conditions or a single impediment from within can create disruptions and generate enough stress to cause the entire system to fail. An attempt to adapt to new environmental conditions would require major system re-development and major operational testing. From a company perspective, this can be costly, time consuming, disruptive and can lead to lost competitiveness in the

marketplace. Instead, modern complex manufacturing environments are moving toward intelligent control systems that are flexible and capable of adapting to stochastic and volatile demand environments [13], [5].

Recognition of the inadequacies inherent in traditional systems has spurred growing interest in alternative control frameworks and attention in the research community has been given to holonic architectures and multi-agent systems [16], [5]. Entities in the holonic control framework exhibit properties of being autonomous and also being collaborative. The autonomy of holons allows them to make decisions without consulting a superior entity and their programmed tendency towards cooperation encourages holons to communicate with one another to mutually develop suitable plans for execution and prevent chaos.

This paper presents specifics of the application of the holonic control framework implemented with agent-based technology to a complex automated order distribution system (ODS) (for more details see [17]). The ODS (Order Distribution System) is a kind of “goods-to-person” technology used in order picking environments. The ODS is designed for instances where there is a relatively small number of SKUs (type of products) as a base (on the order of 1600), fulfilling a high number of orders, and small quantities per pick. In a “goods-to-person” system, labor efficiency is increased since worker travel time is eliminated as the system automatically retrieves product containers (containing the “goods”) from storage, routes and transports them to locations where stationary pickers (operators) pick items from the product containers and sort/consolidate these into many individual orders. This system allows operators to focus on picking and sorting, thus increasing order fulfillment accuracy. In this approach, several orders are fulfilled concurrently. This system exhibits high routing complexity where the product containers are retrieved and routed from the warehouse to multiple paths and locations for order picking activities. The application of holonic manufacturing concepts is accomplished by retro-fitting holonic characteristics in the control system for this ODS system.

## **1.1 Holonic Control Concepts and Agent-based Systems**

In the late 1960s, Arthur Koestler first put forth the term ‘Holon’ during his study of structures of living organisms and complex social systems [10]. Koestler observed that organisms and societies are multi-leveled hierarchies composed of stable intermediate forms or self-contained entities, with some entities (parts) being an element of a larger body (wholes). He noted that these parts and wholes do not have a clear separation and used the term ‘holon’ to refer to self-contained entities or sub-wholes in any level on the hierarchy. An organization of holons was coined a holarchy, which can be represented by a branch structure, where holons form the nodes and the vertical and horizontal branching lines represent a communication network, a multi-level interdependency and a dynamic control structure. In short, holons are semi-autonomous units that function, “(1) as autonomous wholes in supra-ordination to their parts, (2) as dependent parts in sub-

ordination to controls on higher levels, and (3) in co-ordination with their local environment”.

With regard to implementation of the holonic control framework, agent technology is viewed by the research community as an enabling software technology that can be used to implement autonomous and cooperative behaviors in distributed systems [3], [12], [11]. Agents in multi-agent systems (MAS) possess the attributes of autonomy and collaboration essential for constructing social behaviors. For instance, agents are capable of autonomously reacting to changes in their environment and performing goal-directed reasoning. Agents are also able to cooperate, coordinate and negotiate with other participating agents to reach mutual agreements in order to meet their design and larger system-wide objectives. Thus, “multi-agent systems supply both the reasoning techniques necessary to implement the information processing architecture of a holon and the cooperation techniques necessary for holons to interact with other holons and to build holarchies” [2].

To capture the communication needed among agents to allow for mutual agreement and collaboration, the Contract Net-Protocol (CNP) [6], [14]), an auction based bidding mechanism, can be used. CNP is a negotiation protocol which uses various decision-making algorithms, such as priority rules, heuristics, and multi-objective functions, to determine the winning bidder. This CNP based collaborative model is used to handle the task of scheduling and resource assignment and to accomplish the kind of communication necessary for building an holonic architecture into the existing ODS.

The general description of the CNP collaborative work/model is shown in Figure 1. The manager holon acts as a coordinator for aiding cooperation among other holons, called contractors. First, the manager announces bid requests to all participating contractors. Second, the contractors construct bids according to their current status and send bids to the manager. Third, the manager collects all bids and begins evaluating them. Bids may be composed of different factors, which may reflect the holons’ local or system objectives. Depending on the environment, the manager may choose to satisfy different objectives and use different strategies for selecting the best bid for a given situation. Fourth, the Manager’s evaluation leads to awarding the task to the contractor with the best bid. The winning contractor then acts to execute the given task. This given task will be considered as part of the schedule and becomes an obligation for the winning contractor to perform the task. This information is taken into account when the new winning contractor constructs its bids for future tasks.

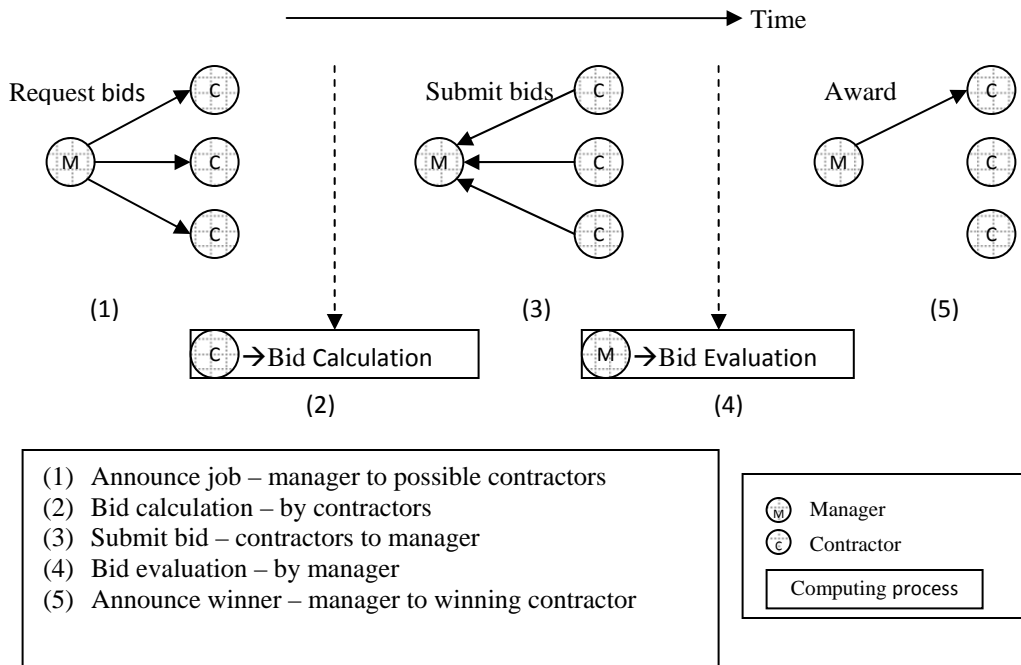


Figure 1: Auction-based Collaboration Model

## 2 Application of Agent-based and Holonic Control Framework

The controls in the existing system are coded in object oriented language, C++, to achieve some flexibility. However, the system also has strong centralized characteristics and retains some of these in the retro-fit. The original codes appear rigid and cannot easily be modified nor adopt new technologies without a redesign of many other interdependent functions. The existing system also relies on creating advanced schedules that often do not reflect the real-time status of the system (therefore schedules tend to be outdated), and the system rigidity leaves it without capability to respond quickly to unexpected disturbances (such as workstation or crane shutdowns or adding new orders).

With aims to resolve these weak points in the existing control system, this paper reports the development of agent-based algorithms for the intelligent real-time scheduling and control of this *goods-to-person* order picking process (the ODS system described in [17]) using holonic concepts and a decentralized control approach.

While a number of the many functions in the existing ODS control system have interlinks and interdependencies on each other, we classify these many functions into three main function groups according to their associated physical entities. These are Order Management, SKU Management, and Crane Management.

The following discusses the application of agent-based and holonic concepts concentrating on the functions of *SKU Management* and *Crane Management*. For the

*Order Management* function group, we leave this topic as part of on-going research work. While summarized in this paper, more details on the Crane Management function group are provided in [17].

## **2.1 Holonic SKU Management Module**

SKU Management serves a crucial purpose in the ODS. It controls the speed with which orders can be fulfilled by managing the flow of SKUs into and within the system. The task of retrieving and routing totes is handled through the *SKU Retrieval (SRet) function and SKU Tote Routing (SRO) functions*. SRet determines when and what SKU tote to retrieve from the warehouse and SRO selects the tote's workstation destination and directs the tote to that destination for order picking activity. It is these two main functions and the following holon entities that compose the Holonic SKU Management processes:

*Global Holon (GH)*: The GH has global knowledge about the system including demand (orders) and supply (SKU totes). By communicating with the Workstation Holons (explained below) and Crane Holons (representing cranes), it is aware of current and expected workload in each workstation and the status of automatic cranes, respectively.

*Tote Router (TR)*: A TR is associated with one product tote that contains items of the same SKU and is responsible for routing the tote to an appropriate workstation where an operator can pick one or more items from the tote and sort into required order containers.

*Workstation Holon (WH)*: A WH is associated with a workstation operator and is knowledgeable about order workstation fulfillment progress and utilization in terms of space management and operator performance. Each workstation is usually assigned several orders and thus contains several Order Holons (OHs). The WH represents these orders by first requesting the most urgent SKUs. A WH learns about orders in a workstation by communicating with the corresponding OHs. The WH also has knowledge about order priorities, order status, and about the number of product totes waiting in its buffer as well as the tote number en route to visit the workstation.

It is the interaction of the above holon entities that allows a system to make decisions and function in real-time. Since the state of the system is dynamic and can change quickly, the process of determining the path for a tote to travel and selecting a workstation destination needs to be sufficiently flexible to cope with constant changes or disturbances. In order to do this effectively, the system must be able to make needed adjustments in real-time. Real-time enabled responses can minimize excess queue time and productivity loss.

Given this observation, the existing system is not equipped with real-time response capability. Instead, it employs a rigid process where the set of workstations to be visited is determined well in advance of execution. This in turn creates a rigidity that impedes the system's ability to respond to disturbances, such as one or more workstations temporarily out of service, and can potentially cause a larger part of the system to shutdown.

In the modified control system, the above holonic entities are introduced to allow for more routing flexibility, and to improve the system's ability to cope with disturbances. We also activate more decision making locations for enhanced rerouting capability thus allowing decisions to be made and implemented more frequently and using real-time or near real-time information. In cases where an unexpected event occurs, the totes' destination plans can change and totes are re-routed to avoid or at least lessen the negative effects that might take place after the disturbance.

### **2.1.1 SKU Routing Function (SRo)**

In the modified ODS, tote schedules are not created in advance. Instead, the TR focuses on scheduling the tote to go to the best workstation by using *an auction-based mechanism among entities* and arriving at a rational and flexible decision among workstations. The brief explanations of the process are as follows.

When a tote with no assigned destination arrives at one of the decision locations in the ODS, the associated Tote Router (TR) sends out bid requests to all active workstations to start the process of finding the best destination for the tote. A bid request includes information about the tote such as the tote ID, the SKU number of the tote, and the quantity contained in the tote, as well as the current location of the tote (will be used for calculating distance between the tote and each workstation). For a tote with an assigned destination, the TR checks whether or not this tote requires rerouting. For example, if the tote has progressed past its next priority location when its original priority workstation destination is shutdown, rerouting will be required. In these cases, the TR is the initiator of the bidding mechanism.

After the TR sends the bid requests, each workstation with capability and capacity develops a bid that is based on its status and the factors that are important to it to satisfy its local goals as well as consistent with system-wide goals. Only active workstations that have demands for the SKU prepare their bids. A space check, however, is also required to prevent routing totes to overloaded lanes and workstations which consequently will create tote traffic in the system. In order for workstations to construct their bids, several real-time (or near real-time) measurement factors are taken into account. These factors are the priority of SKU(x) in workstation (i) ( $Priority_{WS(i)}(x)$ ), workload at the workstation (i) or number of totes waiting for picking in that workstation ( $Workload_{WS(i)}$ ), and the distance between workstation (i) and the tote's location ( $Distance_{WS(i)}$ ).

A formula for the bid calculation for each workstation is as follows:

$$Bid_{WS(i)}(x) = function(Priority_{WS(i)}(x), Distance_{WS(i)}, Workload_{WS(i)}) \quad (1)$$

$$= 100 + Priority_{WS(i)}(x) - Distance_{WS(i)} - j * Workload_{WS(i)} \quad (2)$$

Where

$WS(i)$  is workstation  $i$

$j = 1$  if workload<sub>workstation  $i$</sub>   $\geq 5$  (workload threshold value)

$j = 0$  otherwise

The TR evaluates the bids received and selects the workstation with the best bid as the next destination for the tote. The best bid is the one that has the highest bid value. However, if there is a tie, the bid with highest priority and lowest workload will be chosen.

### 2.1.2 SKU Retrieval (SRet)

A desired characteristic of the SRet function is for it to perform retrieval tasks to fulfill the most urgent demands along with the objective of utilizing all workstations to maximize system performance. The SRet in the ODS is modified to achieve more real time response. It uses demand priority and available space in workstations to determine the next SKU to send into the system. Knowing in real time the demand priority and available workstation space can help to balance workload at workstations. The next best SKU retrieval task can then be performed and waves or cycles of very high and very low utilization in each workstation can be prevented by level loading or balancing workload among workstations.

In SKU Retrieval, the Global Holon (GH) is responsible for choosing the next SKU to be retrieved by the warehouse cranes. GH does this by communicating with workstation holons to aggregate open SKU demand and priority information, including space available at each workstation. The GH also talks to tote router holons to determine the SKU supply in progress. It then calculates which SKU to retrieve using the *open SKU demand*, the demand priority among workstations and the SKU supply-in-progress. The following describes the algorithm for SRet as performed by GH.

Step 1: Compile a list of positive *open SKU demand* across the ODS along with SKU priority and the number of workstations with open demand for the SKU. This is the aggregate SKU list where all demand for the same SKU will be added together and the supply totes containing the same SKU will also be added together. A net open demand quantity for each SKU record is computed.

Step 2: Since the highest demand SKU may come from a workstation that is already full, a space availability check at the workstation needs to be made. Space availability at each workstation is calculated by the anticipated number of product totes at that workstation. This includes the product totes currently located at the workstation and the

product totes destined for this workstation (supply-in-progress). GH selects the SKU with the highest priority and decides whether the SKU can be sent into the system at a particular time. If at least one workstation with that SKU demand has sufficient space, then the GH will generate a new retrieval task for that SKU by directing the crane manager to retrieve a tote containing that SKU. However, if a workstation has insufficient space, then no SKU retrieval task will be generated at that particular time. The SRet will read the next SKU from the list and perform the same calculations. A new list of open SKUs is frequently regenerated when the list expires.

## **2.2 Holonic Crane Management Module**

Crane Management governs the assignment of crane retrieval and crane putaway tasks. The existing Crane control logic is centralized, that is, a top-level scheduler creates schedules for cranes using fixed dispatching rules. In the modified system, crane management is performed through the interaction of the Crane Manager (CM) and Crane Holons (CH), who together, comprise a holonic function group. The CM is a virtual entity responsible for assigning retrieval and put-away tasks to cranes. The CH exists in each crane and has knowledge about assigned tasks and their status, as well as the status of the queue for retrieval and put-away totes. The auction-based collaboration in this holonic function group engages all cranes and produces an agile system for *Crane Retrieval Selection (CRS)* and *Crane Putaway Selection (CPS)*.

The objective of the CRS and CPS functions is to maximize overall crane utilization. In achieving this objective, the CM uses different algorithms to evaluate bids for retrieval tasks and putaway tasks. For retrieval tasks, the CM chooses a crane with the shortest (or smallest) retrieval queue. If more than one crane has the same smallest queue size, the CM will use the next criterion, crane retrieval progress or the number of active retrieval totes, to break the tie. One exception to using the shortest retrieval queue criterion has to do with maximizing the use of each crane's two tote positions. The CM would utilize both positions each time the crane moves into the retrieval zone in order to maximize crane retrieval utilization. For putaway tasks, the CM uses the shortest putaway queue dispatching rule for evaluating bids and choosing an appropriate crane. It uses random selection as a tie breaker

In circumstances when the number of putaway totes starts rising beyond an upper bound threshold, possibly causing congestion and threatening a shutdown in the main loop, the CM's objective will turn to maximizing crane putaway utilization. Therefore, in the modified ODS, the CM sends a request to the GH to interrupt the process of generating retrieval tasks (or SRet) for a designated period of time until the number of putaway totes is reduced to a lower bound threshold. Once the number of putaway totes decreases below this threshold, the CM will request the GH to resume its process of generating retrieval tasks and the CM's strategy will revert back to maximizing overall crane utilization.

### **3 Experiments**

For comparison purposes, the simulation runs were conducted for both the “existing” and “proposed” (also called “modified”) ODS. This study is to obtain results for direct comparisons of the performance of the existing system versus the performance of the modified system and to assess the difference in performance when different order sets are used. This performance comparison is accomplished by running simulations on the unmodified system and the modified system, first under undisturbed condition and then under disturbed conditions (explained in the following section). Three order sets of actual customer orders, with 1,000 orders each, are used in the test runs to ensure that the results are indeed independent of possible unique characteristics in one of the individual test sets.

#### **3.1 Test Conditions**

##### **3.1.1 Non-disturbed Condition**

For the normal condition simulation environments, the entire ODS (which includes the physical system and equipment such as conveyor, workstations, cranes, check point equipment) is expected to run and perform smoothly without any interruptions; there are no system faults introduced in the “normal” run.

##### **3.1.2 Disturbed Condition**

Under disturbance conditions, the simulation run is modified so that, at two different periods, an arbitrary workstation is shut down for some time (see Table 2) and then re-started. Shutting down workstations introduces a stress test for both the modified system and the existing system. Flexibility in recovery and continued high levels of system operation under such fault conditions are attractive characteristics of a control system. Existing orders will not be removed from inoperable workstations and the simulation continues running until all 1,000 orders are completed. Table 1 shows the shut down time for two randomly selected workstations.

Table 1: Workstation shutdown information

<b>Workstation</b>	<b>Uninterrupted Run-time Before Shutdown</b>	<b>Shutdown Duration</b>
2	2 hr	30 min
10	3 hr	30 min

### **3.2 Input order sets characteristics**

The simulation order sets are obtained from an actual industrial order stream composed of 3,355 orders. The total number of different SKUs in the warehouse equals 10,543. Each order is composed of one or more of these SKUs, each composing an order line. Table 2 includes the characteristics of the 3 sets of input order stream data sets used in the experiment. Each order set is comprised of approximately 1,000 orders, leaving a small number of orders from the original order stream unincorporated. There are 10,014 different SKUs represented in the total of the three order sets out of the 10,543 different SKUs in the warehouse.

The order sets were developed by arbitrarily parsing the first, then second, then third 1,000 orders from the entire order stream into the order sets used for testing. The first set contains the lowest average number of orderlines per order at 14.77, with a total order quantity of 44,027 and average quantity per order line of 2.98. The second set contains a total of 15,170 order lines, with a total order quantity of 47,725, giving an average quantity per line of 3.15. The third set has 15,125 order lines and contains the largest of order quantities at 49,716, which averages to a 3.29 order quantity per line.

Table 2: Characteristics of Order Sets used for testing (1,000 orders per set)

<b>Order Info</b>	<b>Order set</b>	<b>Order set</b>	<b>Order set</b>
Total number of distinct SKUs	5,650	6,051	5,739
Total number of orderlines	14,774	15,170	15,125
Average number of orderlines / order	14.77	15.17	15.12
Total order quantity (units or “eaches”)	44,027	47,725	49,716
Average quantity / order (units / order)	44.03	47.73	49.72
Average quantity / orderline (units/orderline)	2.98	3.15	3.29

#### **3.2.1 Performance measure**

The experiment’s performance analysis mainly looks at changes in productivity and cycle time and ignores picking accuracy. Accuracy is ignored since it is affected by system physical design and picking/packing methodology. In this experiment, both the original and modified ODS use the same physical design and picker methodology.

Productivity is measured by the pick rate, i.e. the number of order lines the operator picks per hour (# order lines/hr). In addition, crane utilization is also used as a measure of productivity.

Cycle time performance accounts for the time it takes to get an order from order entry to the shipping dock and is measured as the amount of time it takes to fulfill an entire order in the ODS, i.e. the order processing time. The total time to complete the order stream is used as an additional measurement in comparing the two systems.

## 4 Results & Conclusion

### 4.1 Performance Improvements by Modified ODS

The introduction of holonic concepts at most stages of decision-making results in improved system performance. Under normal conditions, a head-to-head performance comparison of the modified ODS and pre-existing ODS showed the modified ODS had faster average order processing time (22.6-24.3%), improved total time to complete the order stream (22.3-27.8%), better average crane utilization (31.8-40.6%), and higher average operator performance (28.4-38.5%) in all three different order streams under normal conditions.

Under disturbed conditions, the modified ODS showed even greater improvement results. For instance, the improvement of the modified system over the existing system in average order processing time under disturbed conditions ranged from 26.0-29.3%, compared to only 22.6-24.3% in the undisturbed conditions.

The existing system's performance deteriorates significantly under disturbance, while the performance of the modified system only decreases marginally, thus there is a greater performance gap between the modified and existing systems. The better relative performance of the modified over the existing system when disturbance is introduced rather than when no disturbance is present shows its fault tolerance capability. The next section describes the fault tolerance analysis of the existing and modified systems.

Table 3: Performance improvement of the modified ODS compared to the existing ODS under undisturbed condition

Performance measure - Undisturbed condition	% Improvement from Existing ODS		
	order set #1	order set #2	order set #3
Total time to complete order stream	26.7%	22.4%	27.8%
Average order processing time	24.3%	22.6%	23.8%
Standard deviation of order processing time	33.2%	30.9%	28.9%
Average operator performance	35.8%	28.4%	38.5%
Average crane utilization	37.6%	31.8%	40.6%

Table 4: Performance improvement of the modified ODS compared to the existing ODS under disturbed condition

Performance measure - Disturbed condition	% Improvement from Existing ODS		
	order set #1	order set #2	order set #3
Total time to complete order stream	31.0%	24.3%	30.8%
Average order processing time	29.3%	27.3%	26.0%
Standard deviation of order processing time	37.5%	35.7%	31.7%
Average operator performance	46.2%	31.9%	44.8%
Average crane utilization	48.1%	34.2%	47.5%

#### 4.2 Fault Tolerance Analysis of the Existing and Modified ODS

We now look specifically at the degree of fault tolerance across each of the five performance measures. The degree of fault tolerance for each system (the existing system and the modified system) is measured here by % performance deterioration of the system performance under disturbance conditions as compared to the performance under normal condition. The smaller the deterioration, the more fault tolerance the system acquires. Table 5 contains the % of performance deterioration of both the existing ODS and modified ODS for across all three order sets.

Table 5: Percentage of deterioration from disturbances of the existing and modified ODS

Systems	Performance measure	% Performance deterioration by disturbances		
		Order set #1	Order set #2	Order set #3
Existing ODS	Total time to complete order stream	7.6%	8.2%	6.1%
	Average order processing time	8.9%	8.7%	4.3%
	Standard deviation of order processing time	10.4%	8.9%	8.8%
	Average operator performance	7.2%	7.3%	6.0%
	Average crane utilization	6.4%	5.6%	5.3%
Modified ODS	Total time to complete order stream	1.1%	5.4%	1.7%
	Average order processing time	1.7%	2.1%	1.2%
	Standard deviation of order processing time	3.2%	1.4%	4.5%
	Average operator performance	0.1%	4.8%	1.6%
	Average crane utilization	-0.7%	3.9%	0.7%

The % of performance deterioration is calculated, for example, by taking the total processing time of the existing system under undisturbed conditions (normal condition), and comparing that to the total processing time of the same system under disturbance conditions, which calculates to a 7.6% increase in total processing time. This additional increase in time is labeled as % deterioration.

For the existing system, performance deterioration of order stream completion time, across all three order sets, ranges from 6.1% to 8.2%. For the modified ODS, the same results showed only a 1.1% to 5.4% decline in performance. The performance deterioration in the next category is average order processing time. While the existing ODS showed decline of 4.3% to 8.9%, the modified ODS only experienced a decline of 1.2% to 2.1%. The increased variation in order processing time that resulted from the disturbance ranges from 8.8% to 10.4% for the existing ODS; compared to only 1.4% to 4.5% for the modified ODS.

The average operator performance also suffered with the introduction of disturbance. Within the existing system, operator performance was lowered by 6.0% to 7.3%; while the equivalent performance within the modified ODS fell only 0.1% to 4.8%. Similarly, average crane utilization in the existing ODS saw performance dropped by 5.3% to 6.4%, whereas the modified ODS only experienced a decline of 0.0% to 3.9%.

Introducing disturbances affected the crane utilization in both the Existing ODS and the Modified ODS. Significant deterioration of from 5.3% to 6.4% occurred in crane utilization with the existing ODS. Much less effect was observed in crane utilization in the Modified ODS when disturbances were introduced, namely a range from -0.7% (an observed improvement) to 3.9% deterioration. This one experimental result of a 0.7% utilization improvement under disturbances was not considered significant and likely due to random effects in the experimental run.

The results demonstrate greater fault tolerance in the modified system; the control system demonstrated only slight deterioration in performance, as measured by the five performance metrics. These results show that the holonic-based system (modified) recovered rapidly and maintained good performance, while the existing control system experienced a significant decline in performance with cascading effects on subsystem components.

### **4.3 Conclusion & Future Research Directions**

This research explored the retro-fitting of holonic elements into an existing hierarchical control system to improve the performance with system comparisons viewed in terms of more flexibility, adaptability and fault tolerance capabilities. Agent technology was used as an enabling tool for building autonomous, proactive, and goal-oriented entities and for supporting interaction and communication among entities within the existing architecture. The newly modified system was then evaluated, using stress testing, to assess performance improvements in both normal and typical disturbance conditions. These

experiments showed very promising results, in terms of responsiveness, throughput, utilization, and fault tolerance.

We observe that these entities and rules and strategies might be implemented with either hierarchical or holonic architectures. One would expect similar performance improvements over the original control system with either implementation when disturbances are not present. Our experiments did not include an improved hierarchical control system for comparison. Our observations are, however, that disturbances or exception scenarios in such environments, e.g. order changes or operator downtime, are relatively common and the holonic architecture implementation proves out as the better performer.

The existing system hardware was untouched in our experiments except for the activation of more of the existing sensors in the system to provide more decision flexibility. System performance under deterministic test conditions is improved by introducing holonic concepts. System performance under disrupted conditions is markedly improved through the use of holonic conditions. These results argue strongly for the retrofit of control systems with agent based controls based on holonic concepts. Such changes are software changes and most experts [15] concur that such changes are much less expensive to make than those to hardware. Thus we have the basis for also claiming a cost-effective system performance improvement through introduction of holonic concepts. We have not yet explored possible holonic system benefits in areas such as system scalability and system reusability or modularity.

As the holonic concepts in this research are implemented using agent technology, advances in this area will offer valuable tools for future development work of holonic systems. Additionally, future work can be conducted to measure re-configurability and scalability, and to extend this research methodology to test other real world complex logistical systems.

## **References**

- [1] Brennan, R.W., Norrie, D.W. (1998). Evaluating the performance of alternative control architectures for manufacturing. *Proceedings of the IEEE International Symposium on Intelligent Control*, (pp. 90-95)
- [2] Bussman, S. (1998). An Agent-Oriented Architecture for Holonic Manufacturing. *Proceedings of First International Workshop on IMS*, (pp. 1–12).
- [3] Bussman, S., Jennings, N.R., Wooldridge, M. (2005). *Multiagent Systems for Manufacturing Control*. Springer.

- [4] Cavalieri, S., Geretti, M. Macchi, M., Taisch, M. (2000). An experimental benchmarking of two multi-agent architectures for production scheduling and control. *Computers in Industry* , 43 (2), 139-152.
- [5] Council, N. R. (1998). *Visionary Manufacturing Challenges for 2020*. Washington, D.C.: National Academic Press.
- [6] Davis, R., Smith, R. (1983). Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence* , 20 (1), 63-109.
- [7] Heragu, S.S., Graves, R.J., Kim, B.-I., St Onge, A. (2002). Intelligent agent based framework for manufacturing systems control. *Systems, Man and Cybernetics, Part A* , 32 (5), 560-573.
- [8] Kelly, K. (1999). *New Rules for the New Economy: 10 Radical Strategies for a Connected World*. USA: Penguin.
- [9] Kidd, P. (1994). *Agile Manufacturing, Forging New Frontiers*. Addison-Wesley.
- [10] Koestler, A. (1967). *The Ghost in the Machine*. London, UK: Hutchinson & Co.
- [11] Marik, V., Fletcher, M., Pechoucek, M. (2002). Holons & agents: recent developments and mutual impacts. *Multi-Agent Systems and Applications II* , 233-267.
- [12] McFarlane, D. (1995). *Holonic Manufacturing Systems in Continuous Processing: Concepts and Control Requirements*.
- [13] NGM. (1997). *Next Generation Manufacturing - A Framework for Action*. Next Generation Project Report. Agility Forum.
- [14] Smith, R. (1980). The contract net protocol. *IEEE Transactions on Computers* , C-29 (12), 1104-1113.
- [15] St. Onge, Arthur (2007), conversation at MHIA Integrated Systems and Controls meeting, Savannah, Georgia,
- [16] Valckenaers, P., Bonneville, F., Van Brussel, H., Bongaerts, L. and Wyns, J. (1994). Results of the Holonic control system benchmark at K.U. Leuven. *Proceedings of International Conference on Computer Integrated Manufacturing and Automation Technology*, (pp. 128-133).
- [17] Wan, V. K. (2007). *Complex Integrated Order Fulfillment Technology: A Holonic Approach*. Hanover, NH, USA: Ph.D. Dissertation, Dartmouth College.